



LLM with Physics Parameters for CAD/Solid Modelling

BY

BHAVIKA BABURAJ

2023B5A71210H

Under the supervision of

DR. Sk Aziz Ali

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
OF**

PHY F266: STUDY PROJECT

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI
HYDERABAD CAMPUS**

(March 2025)

Acknowledgement

I express my sincere gratitude to Dr. Sk Aziz Ali for his valuable mentorship throughout the course of this project so far. His patience and expertise has helped me overcome several challenges in the project. I would also like to thank my peers who are working under Dr. Sk Aziz Ali for inspiring me and being a pillar of support throughout the project.



**Birla Institute of Technology and Science-Pilani,
Hyderabad Campus**

Certificate

This is to certify that the project report entitled “**LLM with Physics Parameters for CAD/Solid Modeling**” submitted by Ms. Bhavika Baburaj (ID No. 2023B5A71210H), in partial fulfillment of the requirements of the course PHY F266, Study Project Course, embodies the work done by her under my supervision and guidance.

DR. Sk Aziz Ali

Date: 30th April 2025

BITS-Pilani, Hyderabad Campus

Abstract

There are several applications of AI and Machine learning in the field of CAD Modelling. They range from automating product-design to experimenting with the physics of material science and structural properties for the best results in developing infrastructure and building durable materials for various applications. The aim of this project is to construct a model, which, given a natural language description of the Boundary Representation(B-Rep) of a CAD model, is able to reconstruct the BRep. The methodology used involves reconstruction of the model from a parametric construction sequence of the BRep. For the data annotation pipeline, GPT-4o has been implemented to generate text prompts. The annotations are prepared using the DeepCAD dataset, ensuring high-quality training data for the model.

Contents

Title Page	1
Acknowledgement	2
Certificate	3
Abstract	4
Introduction	6
Data Annotation Pipeline	7
Conclusion	12
Conclusion	14
References	15

Introduction

Computer Aided Design(CAD) has revolutionized the design and engineering industry by digitally aiding the process of design to become faster and more efficient. Yet, in today’s world it still has its limitations as the process of design is slow and laborious. Integrating AI(Artificial Intelligence) and ML(Machine learning) into the design process can solve this problem. It will enable designers to focus on creative innovations rather than repetitive tasks that take up a lot of time in the design process.

Boundary Representation(B-Rep) is a fundamental representation in CAD Modeling. It defines a model through a parametric representation using faces, vertices, edges, and topology of the model. It is a preferred choice of representation for models that need precise measurements, and it offers more flexibility in changing the shape of the model through parametric adjustments. B-Rep plays an important role in modeling physics phenomena and experiments in structural and material sciences, as precise measurements and representations are required for stress distributions, fluid dynamics, and thermal properties.

To contribute to bridging the gap between precise modeling of physical phenomena and AI-assisted design, this project focuses on designing a model which, given a natural language description of the model as input, will produce the B-Rep of the model described. The methodology involves producing parametric construction sequences and converting them into the final B-Rep to ensure geometric accuracy and easy editability.

A key challenge to proceed further is to construct a high-quality dataset with text annotations for models mapped to their B-Rep construction sequences. To tackle this, earlier I used GPT-4o to generate the text prompts. Now, I have implemented an improved automated data annotation pipeline that uses local LLM to produce text prompts from a JSON vector input containing the details of parameters of faces, edges, vertices, and the overall topology of the B-Rep. This method helps create clear and consistent text descriptions from geometry data, making it easier to train models that link natural language with B-Rep construction steps.

This model, if integrated into CAD, can open new doors for physics-intensive fields like aerospace engineering and material sciences to automate parts of prototyping and increase the accuracy of physics-based simulations. In this report, I present the automated annotation pipeline for the DeepCAD dataset and the proposed model architecture.

Data Annotation Pipeline

This project uses the DeepCAD dataset, which has 179,133 models of moderate complexity. The data annotation pipeline consists of two main steps.

The first step involves generating textual descriptions for the models in the dataset. For this, we use a part of the code for BRep Extractor from *BRep Boundary and Junction Detection for CAD Reverse Engineering (IEEE ICM124)* to extract JSON vectors that encode the parameters defining faces, vertices, edges, and overall topology of the B-Rep using topological walks. These JSON vectors are then passed to the local LLM through a custom data loader I developed. The automated annotation pipeline was executed on HPC with H100 GPUs to generate the text descriptions. The resulting annotations align well with the geometric features of the models, effectively completing the dataset labeling process.

Apart from the parameter vectors, for solid modeling, I'm also using physics-based parameters such as volume, surface area, packing fraction, porosity and surface-area-to-volume ratio to describe the solid model with accuracy. These geometric properties of the solid play a huge role in manufacturing, design, and material science industries. Volume is the total 3D space occupied by the model and it plays a huge role in calculating the mass of the solid via its density to estimate material quantities required in the industry. Surface area is the total area of the object's exterior and is very important in processes like coating and finishing, surface treatments, and during heat transfer and fluid dynamics. The packing fraction measures the efficiency of the space utilized and is important in additive manufacturing. Porosity is the ratio of the void volume to the total volume which is crucial to understand lightweight and cellular structures. Finally, the surface area-to-volume ratio shows the relationship between an object's surface area and its volume, where a higher ratio indicates greater surface exposure, which is important for applications like catalysis, cooling systems, and balancing strength-to-weight considerations.

```

1  [INST]
2  You are a senior CAD engineer. The following model has:
3  - a total volume of 0.785 units³
4  - surface area of 3.14 units²
5  - packing fraction: 0.89
6  - surface-to-volume ratio: 4.0
7  - center of mass at (0, 0, 0.5)
8  - moment of inertia: [0.1, 0.1, 0.2]
9
10 Generate a natural language instruction to describe this shape in terms of both geometry and physical behavior.
11
12 [/INST]

```

Figure 2: Prompt with Physics Parameters

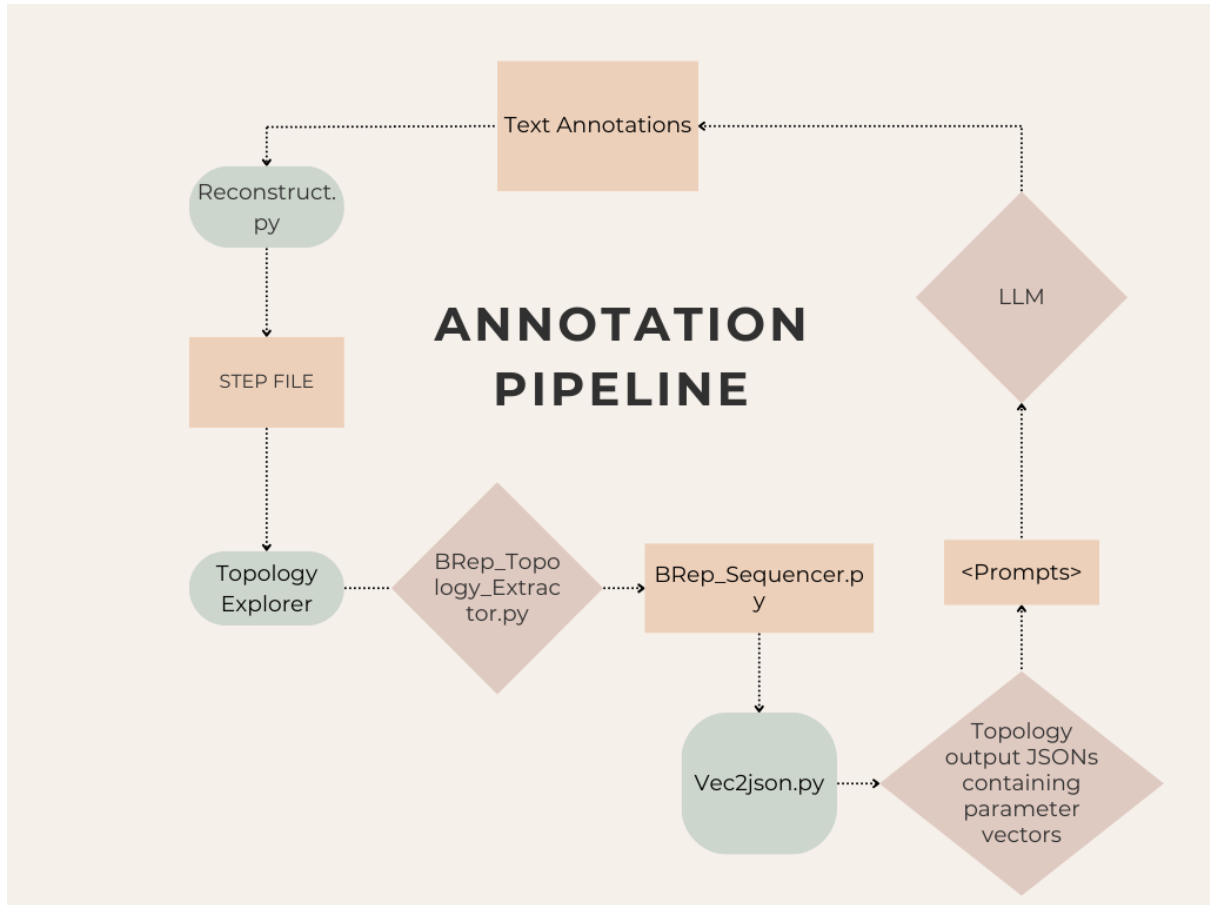


Figure 1: Data Annotation Pipeline Flowchart

```

1  {
2    "primitive_type": "cylinder",
3    "dimensions": [0.5, 0.5, 1.0],
4    "translation": [0.0, 0.0, 0.0],
5    "rotation": [0, 0, 0, 1],
6    "volume": 0.785,
7    "surface_area": 3.14,
8    "packing_fraction": 0.89,
9    "sv_ratio": 4.0,
10   "center_of_mass": [0.0, 0.0, 0.5],
11   "inertia": [0.1, 0.1, 0.2]
12 }

```

Figure 3: JSON output Parameter vector with physics parameters

The second step involves generating BRep sequence vectors inspired by the methodology in *BRep2Seq: A Dataset and Hierarchical Deep Learning Network for Reconstruction*. In BRep2Seq, the construction sequence is represented as a vector of operations, which include principal primitives and detailed features. In this report, we focus only on generating the principal primitive vector. Each principal primitive is represented by a parameter vector:

$$[l_1, l_2, l_3, e, t_x, t_y, t_z, q_0, q_1, q_2, q_3]$$

where:

- $l_1, l_2, l_3 \rightarrow$ Length, breadth, and height of the principal primitive.
- $e \rightarrow$ Extrusion width.
- $t_x, t_y, t_z \rightarrow$ Translation of the principal primitive relative to the global coordinate system, representing the position of its center.
- $q_0, q_1, q_2, q_3 \rightarrow$ Rotation transformation of the principal primitive.
- Unused parameters are set to -1.

The exact values of these parameters are extracted from the JSON parameter vectors created in the previous step. Consequently, each model now has three corresponding text prompts (beginner, moderate, advanced), which are mapped to their respective principal parameter vectors.

Additionally, a sanity check is performed to ensure the correctness of these parameter vectors by comparing them with actual model dimensions in Onshape (Figure 2,3). Figure 4(referenced from BRep2Seq) illustrate the principal parameter vector in detail.

Table 1: Modeling operations and parameters of principal primitives.

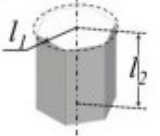
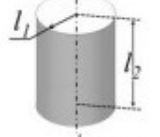
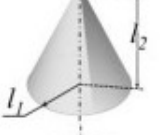
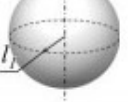
Operation	Parameters	Description	Visualization
B(Box)	$l_1, l_2, l_3, t_x, t_y, t_z, q_0, q_1, q_2, q_3$	Cuboid defined by length (l_1), width (l_2), and height (l_3).	
P(Prism)	$l_1, l_2, e, t_x, t_y, t_z, q_0, q_1, q_2, q_3$	e-Sides prism defined by radius (l_1) and height (l_2).	
C(Cylinder)	$l_1, l_2, t_x, t_y, t_z, q_0, q_1, q_2, q_3$	Cylinder defined by radius (l_1) and height (l_2).	
K(Cone)	$l_1, l_2, t_x, t_y, t_z, q_0, q_1, q_2, q_3$	Cone defined by radius (l_1) and height (l_2).	
S(Sphere)	l_1, t_x, t_y, t_z	Sphere defined by radius (l_1).	

Figure 4: Principal primitives and their parameter vectors. Source: BRep2Seq [?].

[Text2BRep/BRep_Sequencer_Output_Samples/00003_index_3_output.json](#)
Lines 32 to 43 in d3ca80b

```

32   "feature_type": "cylinder",
33   "param": {
34     "L1": 0.4839,
35     "L2": 0.9697,
36     "T_x": -0.2249,
37     "T_y": 0.0,
38     "T_z": 0.0,
39     "Q_0": 1.0,
40     "Q_1": 0.0,
41     "Q_2": 0.0,
42     "Q_3": 0.0
43   }

```

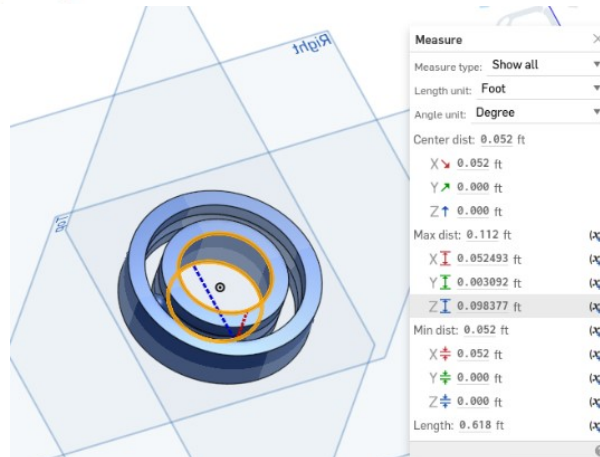


Figure 5: Z component of length 0.098377 matches with "L2": 0.9697 ($\times 10^{-1}$)

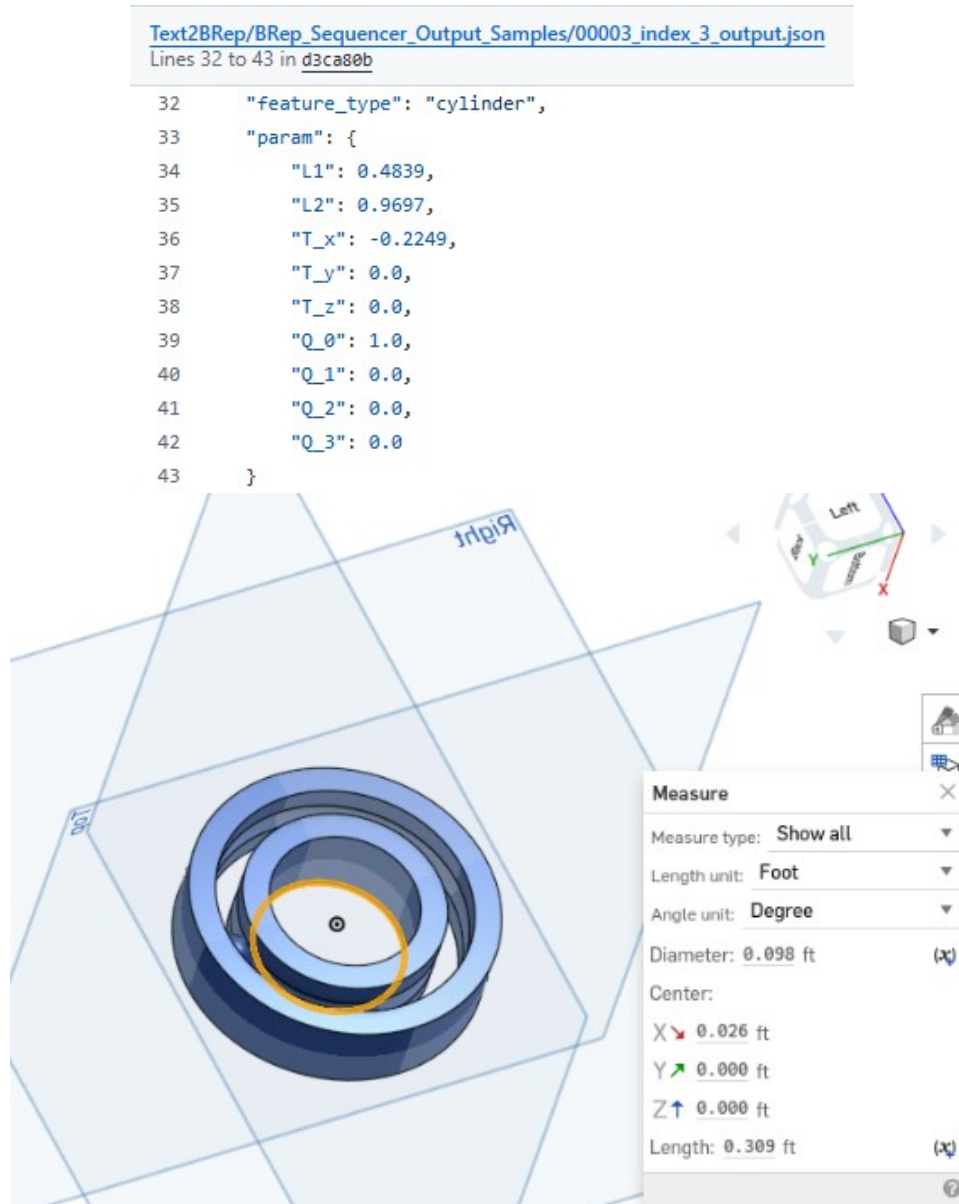


Figure 6: Diameter = 0.098 ft means radius = 0.049 ft. This matches with "L1": 0.4839 ($\times 10^{-1}$ ft)

Model Architecture

My proposal for the model architecture is nearly identical to what is used in *Text2CAD*. It is a transformer based architecture, that uses a Pre-trained BERT ENcode to generate contextual embeddings.

Text Encoder Module

The input text prompt $T \in R^{N_t}$ is tokenized and passed through a pre-trained BERT model to generate contextual embeddings:

$$\mathbf{T} = BERT_{pre-trained}(T)$$

Here, $\mathbf{T} \in R^{N_p \times d_p}$, where:

- N_p is the number of tokens in the input text.
- d_p is the dimensionality of each token embedding.

To adapt these embeddings to the CAD domain, an adaptive transformer layer is introduced, which consists of a single transformer encoder layer. This layer refines the output embeddings $\mathbf{T}$ to align better with the domain-specific vocabulary and structural semantics required for CAD applications.

CAD Token Representation

Each CAD token is represented as a structured tensor, not a flat vector. It comprises two components:

- **Operator Vector** ($\mathbf{o}_t$): A one-hot encoded vector of shape (32,) that specifies the type of CAD primitive or geometric feature (e.g., box, cone, cylinder).
- **Parameter Matrix** ($\mathbf{p}_t$): A matrix of shape (11, 256), where:
 - The number 11 corresponds to the geometric parameters associated with the operator (e.g., length, radius, etc.).
 - Each parameter is quantized into 256 discrete bins and one-hot encoded, yielding a 256-dimensional vector per parameter.

Thus, a single CAD token is represented by the structured tensor:

$$CADToken = \mathbf{o}_t \oplus \mathbf{p}_t = (32,) \oplus (11, 256) \Rightarrow Tensorshape : (32, 11, 256)$$

Full CAD Sequence Representation

To represent the entire dataset of CAD sequences, we define the following:

- N : The batch size, i.e., the number of sequences in a batch.
- T : The sequence length, i.e., the number of tokens (CAD primitives or features) per sequence.

Each CAD token in a sequence has a structured representation with shape $(32, 11, 256)$, capturing both the operator type and the quantized geometric parameters. Therefore, the full CAD dataset is represented as a 5D tensor:

$$\mathbf{O} \in R^{N \times T \times 32 \times 11 \times 256}$$

This representation preserves the symbolic and parametric structure of the CAD tokens across the batch and sequence dimensions, enabling the model to learn and generate coherent and valid CAD geometries.

CAD Sequence Modeling and Training

We can pass the structured CAD sequence tensor to the transformer-based decoder architecture that will autoregressively generate BRep tokens based on the encoded text representation. The decoder will utilize a **cross-attention mechanism** to look over the output of the text encoder (from BERT and the adaptive layer), and align the textual description and the geometric representation.

At each decoding step, the transformer predicts the next BRep token by computing the probability distribution over the operator and parameter vectors, conditioned on previously generated tokens and the text prompt. The model can be trained end-to-end using the **cross-entropy loss** between the predicted token distribution and the ground truth CAD tokens:

$$\mathcal{L}_{CE} = - \sum_{t=1}^T \log P(c_t \mid c_{1:t-1}, T; \theta)$$

where θ are the learnable parameters of the model, c_t is the target token at timestep t , and T is the text input.

This formulation allows the model to learn both the symbolic structure and the parametric detail of CAD sequences in alignment with textual descriptions.

Conclusion

This project lays the groundwork for further advancements in AI applications in CAD model reconstruction and is a step further in increasing the efficiency of the design process. By implementing a structured data annotation pipeline, i.e, using GPT-4o to produce text prompts and creating principal parameter vectors which form the BRep construction sequence for each sample, we ensure high-quality input for training the model. Future work will focus on designing the model architecture, optimizing it, working on the reconstruction process and careful evaluation of the model's performance.

References

1. **BRep Boundary and Junction Detection for CAD Reverse Engineering** Authors: Zhang, Y., Liu, W. Year: 2024
2. **DeepCAD: A Deep Generative Network for Computer-Aided Design Models** Authors: Wu, H., Zhang, H., Cohen-Or, D. Year: 2021
3. **CADParser: A Parser for Computer-Aided Design Models** Authors: Li, X., Wang, Y. Year: 2023 Source: Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI-23), 200-207.
4. **BRepGAT: Boundary Representation Data Augmentation for Geometric Analysis Tasks** Authors: Chen, J., Xu, K. Year: 2023 Source: Proceedings of the 2023 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 106-115.
5. **SkexGen: Sketch-Based Generation of 3D Shapes** Authors: Park, J., Lee, S. Year: 2022
6. **CAD-SIGNet: CAD Language Inference from Point Clouds using Layer-wise Sketch Abstraction** Authors: Khan, M. S., Sinha, S., Sheikh, T. U., Stricker, D., Ali, S. A. Year: 2024 Source: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, 1234-1243.
7. **Text2CAD: Generating Sequential CAD Models from Beginner-to-Expert Level Text Prompts** Authors: Khan, M. S., Sinha, S., Sheikh, T. U., Stricker, D., Afzal, M. Z. Year: 2024
8. **BRepNet: A Topological Message Passing System for Solid Models** Authors: Willis, K. D. D., Ganter, M. Year: 2021
9. **BRep-BERT: Pre-training Boundary Representation BERT with Sub-graph Node Contrastive Learning** Authors: Lou, Y., Li, X., Chen, H., Zhou, X. Year: 2023 Source: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23).
10. **Brep2Seq: A Dataset and Hierarchical Deep Learning Network for Reconstruction and Generation of Computer-Aided Design Models** Authors: Zhang, S., Guan, Y., Chen, H. Year: 2024 Source: Journal of Computational Design and Engineering, 11(1), 110–123.
11. **Design of Porous Structures** Authors: M. Naveen Kumar, B.Gurumoorthy Year: 2021 Source: *Computer-Aided Design and Applications*, Vol. 18, No. 4, pp. 704–713.